Homework 1

Below are four faulty programs. Each includes test inputs that result in failure.
Answer the following questions about each program.

```
/**
 * Find last index of element
 *
 * @param {Object[]} x - The array to search.
 * @param {number} y - The value to look for.
 *
 * @returns {number} Last index of y in x; -1 if absent.
 * @throws TypeError if x is not an array or y is not a
 *         number.
 */
function findLast(x, y) {
    if (!Array.isArray(x)) {
        throw new TypeError('The first parameter must be
an array');
    }
    if (typeof y !== 'number') {
        throw new TypeError('The second parameter must be
a number');
    }
    for (let i = x.length - 1; i > 0; i--) {
        if (x[i] === y) {
            return i;
        }
    }
    return -1;
}
// test: x = [2, 3, 5]; y = 2; Expected = 0
```

```
/**
 * Find last index of zero
 *
 * @param {Object[]} x - The array to search.
 *
 * @returns {number} Last index of 0 in x; -1 if absent.
 * @throws TypeError if x is not an array.
 */
function lastZero(x) {
    if (!Array.isArray(x)) {
        throw new TypeError('Not an array');
    }
    for (let i = 0; i < x.length; i++) {
        if (x[i] === 0) {
            return i;
        }
    }
    return -1;
}
// test: x = [0, 1, 0]; Expected = 2
```

```
/**
 * Count positive elements
 *
 * @param {Object[]} x - The array to search.
 *
 * @returns {number} Count of positive elements in x.
 * @throws TypeError if x is not an array.
 */
function countPositive(x) {
    if (!Array.isArray(x)) {
        throw new TypeError('Not an array');
    }
    let count = 0;
    for (let i = 0; i < x.length; i++) {
        if (x[i] >= 0) {
            count++;
        }
    }
    return count;
}
// test: x = [-4, 2, 0, 2]; Expcted = 2
```

```
/**
 * Count odd or postive elements
 *
 * @param {Object[]} x - The array to search.
 *
 * @return {number} Count of odd/positive values in x.
 * @throws TypeError if x is not an array.
 */
function oddOrPos(x) {
    if (!Array.isArray(x)) {
        throw new TypeError('Not an array');
    }
    let count = 0;
    for (let i = 0; i < x.length; i++) {
        if (x[i] % 2 === 1 || x[i] > 0) {
            count++;
        }
    }
    return count;
}
// test: x = [-3, -2, 0, 1, 4]; Expected = 3
```

(a) Explain what is wrong with the given code.
    Describe the fault precisely by proposing a modification to the code.
(b) If possible, give a test case that does not execute the fault.
    If not, briefly explain why not. (You need to give the same number of arguments.)
(c) If possible, give a test case that executes the fault, but does not result in an error state.
    If not, briefly explain why not. (You also need to answer expected and actual output.)
(d) If possible, give a test case that results in an error state, but not a failure.
    If not, briefly explain why not. (You also need to answer expected and actual output.)
(e) For the given test case in (d), describe the first error state. Be sure to describe the complete state.

Hints:

Fault:	Error State:	Failure:
* A static defect in software. * Parts of source code that are incorrect.	* An incorrect internal state. * State information: variable values (including return value).	* External, incorrect behavior with respect to the expected behavior.